

Defensive complexity in antagonistic coevolution

Erick Chastain

Department of Computer Science, Rutgers University, New Brunswick, NJ

Rustom Antia

Department of Biology, Emory University, Atlanta, GA

Carl T. Bergstrom

*Department of Biology, University of Washington, Seattle, WA 98195-1800 and
Santa Fe Institute, Santa Fe, NM*

One strategy for winning a coevolutionary struggle is to evolve rapidly. Most of the literature on host-pathogen coevolution focuses on this phenomenon, and looks for consequent evidence of coevolutionary arms races. An alternative strategy, less often considered in the literature, is to deter rapid evolutionary change by the opponent. To study how this can be done, we construct an evolutionary game between a controller that must process information, and an adversary that can tamper with this information processing. In this game, a species can foil its antagonist by processing information in a way that is hard for the antagonist to manipulate. We show that the structure of the information processing system induces a fitness landscape on which the adversary population evolves, and that complex processing logic is required to make that landscape rugged. Drawing on the rich literature concerning rates of evolution on rugged landscapes, we show how a species can slow adaptive evolution in the adversary population. We suggest that this type of defensive complexity on the part of the vertebrate adaptive immune system may be an important element of coevolutionary dynamics between pathogens and their vertebrate hosts.

I. INTRODUCTION

Coevolution is often antagonistic, such that one species benefits from the other's loss. Classic examples include predators and their prey, and pathogens and their hosts. Antagonistic coevolution is commonly thought to result in rapid co-evolutionary arms races (1). When participants in a coevolutionary arms race can tamper with their opponents' control systems, as microbial pathogens do with host immune regulation (2, 3), we might expect to see a series of subversion efforts and subsequent countermeasures deployed over evolutionary time. Thus one might expect rapid evolutionary divergence in the systems involved in controlling and regulating the attacks and defenses used in antagonistic interactions (4).

However, antagonistic coevolution need not always lead to rapid evolutionary change. Mechanisms that prevent subversion can halt coevolutionary arms races; the field of cryptography abounds with examples of such systems. In a prescient 1955 letter only recently declassified, John Nash anticipated this result (5):

...for almost all sufficiently complex types of enciphering, especially where the instructions given by different portions of the key interact complexly with each other in the determination of their ultimate effects on the enciphering, the mean key computation length increases exponentially with the length of the key....As ciphers become more sophisticated the game of cipher breaking by skilled teams, etc., should become a thing of the past.

Nash was right; one important example is the RSA

cryptosystem (6), in which two parties' communication over a network cannot be decoded by adversaries, unless they successfully find the prime factors of a large number. Prime factorization has been proven to be computationally difficult (or in the parlance of computer science, has high time complexity) and so the system is effectively secure. The main insight for RSA was that a mechanism can be made secure against subversion by using intractability or complexity as a defense. In this paper we explore how *defensive complexity* strategies can be generalized to domains beyond cryptography — for example, immunology.

To explore the role of defensive complexity in antagonistic coevolution, we introduce a new evolutionary game, the *control network game*. This game features two players, the *controller* and the *adversary*. The controller aims to respond appropriately to the state of the environment. To do this, the controller deploys a control system intermediating between sensors that receive a cue about the state of the world, and effectors that take an action. This control system is codified as a control logic with the cue as input and the effector responses as outputs. The controller's payoff is a function of the world state and the actions taken. The adversary aims to interfere, and can do so by tampering with some of the signals in the control logic.

We study what happens when this game is played in an evolutionary context. We pay particular attention to the case in which the controller must first deploy a control logic, and the adversary then has multiple periods in which to learn how to manipulate it. Such a state of affairs could come about for many reasons. One common biological scenario is when learning occurs at the popula-

tion level by the mechanism of evolution by natural selection and evolutionary rates differ, as is the case in a vertebrate host deploying an immune control system against rapidly evolving pathogen adversaries (7, 8). This scenario provides us a well defined and formally tractable learning system to study, in which an asexual adversary population evolves on a fitness landscape and the relevant phenotype space corresponds the set of possible manipulations to the control system. Questions of which control systems are hard to learn reduce to questions about the rate of evolution given the fitness landscape induced for the adversary by the control system. We can then use population genetic analysis to characterize rates of evolution for typical classes of fitness functions. We show that complex control networks can generate sign epistasis in this fitness landscape, thereby carving fitness valleys that must be crossed and reducing the rate at which an antagonist population evolves to subvert the network. Thus control systems afford defensive complexity against natural adversaries if they induce fitness functions for attempted manipulation that take the form of rugged adaptive landscapes with long deep fitness valleys. Where sufficient defensive complexity is in place, antagonistic co-evolution can lead to long periods of structural stasis instead of rapid change driven by an ongoing arms race.

II. DEFENSIVE COMPLEXITY OF SIGNALING NETWORKS

To exploit a signaling system, an adversary must (1) construct or disrupt signals used in the system, and (2) do so in a way that increases its own fitness. In host-pathogen interactions, step 1 is often simple. For example, viruses readily perturb the cytokine signaling network used by the host, by altering gene expression or by producing cytokine mimics and antagonists (9, 10). The latter problem—manipulating the signals in advantageous ways—may be much harder. This is the challenge we focus on here. To do so, we make the *universal construction assumption*: the adversary can construct any signal, but does not know what the signals do. Making the universal construction assumption allows us to reformulate the problem of evolving to manipulate the host’s control network as a learning problem. Defensive complexity then reduces to non-learnability of the control system by the adversary population.

A. Control Network Games

We consider the two-player game between controller and adversary illustrated in figure 1A. In each instance of the game, a cue contains information about the state of the environment. The controller aims to transduce this cue into an appropriate response. To do so, the controller’s sensory apparatus detects the cue, and produces internal signals that will trigger the controller’s response. The response is determined by a control logic tuned to

accomplish some task T . This control logic is selected from the set $L(T)$ of minimal-cost control logics for this task, i.e., from a set of control logics that all perform optimally on the task T . The adversary aims to alter the controller’s response and does so by perturbing the controller’s internal signals.

We explore a game in which the control logic operates on two signals S_1 and S_2 . We model the control logic as a simple branching logic on the signals as described in the Methods. A fitness landscape is induced by the example control network as follows: the adversaries attempt to tamper by up- or down-regulating the signals S_1 or S_2 . Up-regulation or down-regulation of the signals cause changes to the response value R_A . For modeling purposes we only need to describe the control logic and its fitness consequences on the adversary as a function of perturbations to the signals. We can do so in a way that is equivalent to using a particular family of control logics: in the Methods we show the correspondence between control logics from this family and control logics with inputs described in terms of perturbations to the signals.

We outline two simple examples of control logics and the fitness landscapes that they induce. Let the fitness of the adversary be 1 when there is no perturbation to the control logic. First consider a simple signaling network that has the control logic L_s and induces a fitness landscape for the adversary as given by Figure 1b. This fitness landscape is a slope, with each step toward down-regulating S_1 and up-regulating S_2 being progressively more beneficial for the adversary. Second, contrast this with a control logic L_c that is more complex in that it requires more logical operations per conditional IF instruction. The control logic L_c generates a multi-peaked fitness landscape as shown in Figure 1c. On this landscape, any perturbation of a single signal away from the starting state will decrease the adversary’s fitness.

B. Rate of learning

In the case of the simple control logic L_s , an evolving adversary population can readily traverse the monotonic fitness landscape shown in Figure 1b. Even when mutation supply is limiting, an adversary population can reach its fitness optimum by fixing two beneficial mutations in succession. This is a relatively fast process, since drift or mutation-selection balance is not necessary to maintain the single mutation upon which the double mutant can arise. We can estimate the timescale on which double fixation occurs. Sequential fixation of two beneficial mutations takes approximately $\tau_s = 1/\sqrt{\mu^2 N}$ generations on average (11), where N is the population size and μ is the mutation rate for each mutation.

The complex fitness landscape induced by L_c and shown in Figure 1c has five peaks. One of these (ΔS_1 down, ΔS_2 up) is a global maximum; the others, including the no-perturbation starting point, are local max-

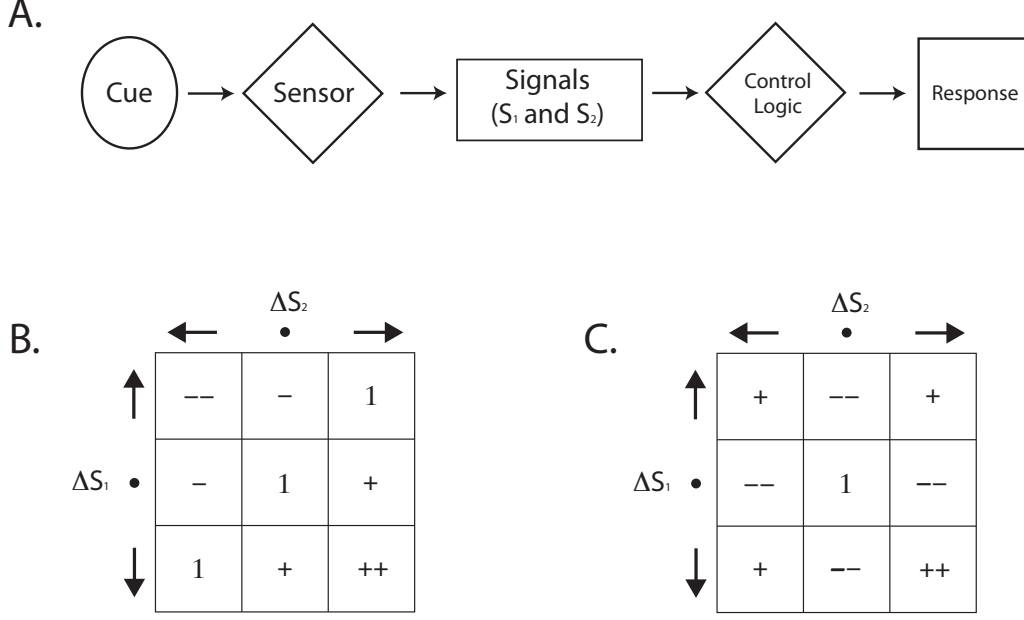


FIG. 1 **Control network game.** A. The controller and adversary play the following game: the controller deploys a control logic and the adversary tries to subvert it. The controller has a sensor that transduces the cue into signals S_1 and S_2 . The control logic then processes these signals to determine the response. The payoff to the controller is maximized when the signals are unperturbed, whereas the fitness of the adversary is some arbitrary function of the responses. Note that this game is thus generically non-zero-sum. The adversary cannot control either the cue or the response directly, but can manipulate the response by tampering with the signals S_1 and S_2 . In our game, the adversary can upregulate, downregulate, or leave each signal unchanged. B. A simple control logic induces a single-peaked fitness landscape on which the adversary can easily evolve to the global optimum. C. A more complex control logic can induce a multi-peaked fitness landscape that prevents the adversary from rapidly evolving to the global optimum.

ima. Between these peaks are fitness valleys that the adversary population must cross. Suppose that the deleterious intermediate in the fitness valley suffers a fitness disadvantage of δ relative to the no-perturbation state. Following Weissman *et al.* (12), when the population size of the adversary is less than $N = \frac{1}{\delta} \log \frac{\delta^2}{\mu s}$ and the selective disadvantage greater than $2\sqrt{\mu s}$, the fitness valley will tend to be crossed by a process known as stochastic sequential fixation. In this process, the deleterious intermediate initially drifts to fixation against selection, and subsequently the beneficial double mutant arises and is fixed by selection. In that regime, the expected number of generations to cross the fitness valley is approximately of order $\tau_c = \mu^{-1} e^{(N-1)\delta}$. When population size of the adversary is larger than $\frac{2s}{\pi\mu^2}$ and selective disadvantage is lower than s , the system lies in the so-called deterministic regime. In this case, the population is large enough that a double mutants are created immediately and these go to fixation. In the deterministic regime the expected number of generations to cross the fitness valley is approximately $\tau_{c'} = \log \left(\frac{s+\delta}{N\mu^2} \right)$.

On the simple landscape induced by L_s , the time to fixation τ_s is of order $1/\sqrt{N}$. On the complex landscape in the stochastic sequential fixation regime, the time to

fixation τ_c is exponential in N . Clearly as N gets large, $\tau_c \gg \tau_s$. In other words, it takes far longer for an adversary population to evolve to manipulate the complex control logic L_c than it does to evolve to manipulate the simple logic L_s . Note, however, that if the adversary population were somehow to become large enough so as to enter the deterministic regime, this result could in principle reverse. For extremely large adversary populations with low cost of ineffective tampering the complex logic can be subverted more quickly than the simple logic, since in the deterministic regime the expected time to fixation τ'_c is of order $\log(1/N)$. The complex network in that case admits an exponential speedup in terms of learning time compared to the simple one.

III. WHAT MAKES A CONTROL NETWORK LEARNABLE?

We have analyzed two example networks that differ in their learnability. The next step is to develop a general theory relating the properties of a control logic to the learnability of that system.

A control logic can be described as a set F of functions—one for each conditional in the branching

logic. Each of these functions f in F takes some input and returns either a 1 (if the formula specifying the conditional evaluates to True) or 0 (otherwise, see Methods). In the example of section 2, the control logic would take the signals S_1 and S_2 as inputs, and determine the appropriate response by evaluating each conditional. We can quantify circuit complexity as follows. The circuit complexity of branch f of the control logic is simply the minimum number of ternary logic gates (see Methods) needed to implement f (13). The circuit complexity of the full logic F is the maximum circuit complexity over f in F .

We consider a control logic to be effectively unlearnable by natural selection if the learning time for this logic is exponential in the number of signals n . In this case, the controller can force the learning time to blow up exponentially by adding even a modest number of signals. The major result of this section is that one can construct an effectively unlearnable control logic with circuit complexity that is linear in the number of signals. Formally,

Theorem 1 *There exists a control logic on n signals with circuit complexity $O(n)$ is learnable in a number of generations exponential in n .*

We prove this theorem in the Appendix; the basic intuition is as follows. Think of the n -dimensional $3 \times 3 \times 3 \dots \times 3$ hypercube where each dimension represents perturbations (down, none, up) to one signal. We establish a control logic by which all corners of the hypercube are global optima and the center is a local optimum. All other spaces are fitness valleys. In other words, we construct a control logic in which global maxima occur only where each and every signal has been altered from its default value. To reach a global optimum, an adversary needs to tamper with n different signals. Then we show that from the starting place where signals are left unperturbed, the expected number of generations needed to produce one of these beneficial n -mutants is exponential in n .

This construction is just one simple example. More complicated fitness landscapes could lead pathogen populations on detours through a sequence of local maxima, delaying convergence to the global maximum.

IV. DISCUSSION

We were motivated by considering antagonistic coevolution such as that between pathogens and the adaptive immune system of vertebrates. Our framework suggests that the kinds of signaling networks present in the immune system induce a complex fitness landscape with valleys and local maxima for pathogens attempting to deceive the immune system. Our results are consistent with two broad observations pertaining to immunology. First, mice serve as a surprisingly faithful model system for uncovering principles of immunological signaling and

control that are also valid in humans even though these two species diverged 75 million years ago. This relative stasis is consistent with the predictions of the defensive complexity model; a sufficiently complex immune circuitry would limit the extent to which rapidly evolving viral and bacterial pathogens provoke coevolutionary arms races than in turn would drive divergence between the immune systems of mice and men. Second, immunologists have found it considerably difficult to decipher the rules behind the functioning of the immune system. Defensive complexity might be expected to give involve to complex rules which are difficult for pathogens to exploit and immunologists to understand. The next step will be to quantify the extent to which these observations arise from defensive complexity rather than other factors.

V. METHODS

We implement the control logics for our example networks using Kleene’s three-valued logic; for truth tables in Kleene’s logic, see ref. (14). We denote the perturbations to signal i with ΔS_i . Each perturbation can take on values $+$, $-$ or \bullet . When instantiating the control logic, we interpret $+$ as 1, $-$ as 0, and \bullet as an input being absent or unknown. For example, in the \neg operation, when given \bullet as input, the output is also \bullet . The AND operator will evaluate to True only if both its inputs are True. The OR operator will evaluate to True only if at least one of its inputs is True. After the branching program terminates—which happens when a particular conditional evaluates to True—then the rest of the conditionals cannot be triggered, and so the order of specification for the conditionals matters a great deal.

We use the notation \wedge for the AND logic gate, we use \vee for the OR logic gate, \oplus for XOR, and we use \neg for NOT. We can implement the control logic L_s as follows:

1. IF $\Delta S_1 \wedge \neg \Delta S_2$, $\Delta R_A = +2m$
2. IF $\Delta S_1 \oplus \neg \Delta S_2$, $\Delta R_A = +m$
3. IF $\Delta S_2 \wedge \neg \Delta S_1$, $\Delta R_A = -2m$
4. IF $\Delta S_2 \oplus \neg \Delta S_1$, $\Delta R_A = -m$
5. ELSE $\Delta R_A = +0$

We can implement control logic L_c as follows:

1. IF $\Delta S_2 \wedge \neg \Delta S_1$, $\Delta R_A = -2m$
2. IF $(\Delta S_1 \wedge \Delta S_2) \vee (\Delta S_1 \wedge \neg \Delta S_2) \vee (\neg \Delta S_1 \wedge \neg \Delta S_2)$, $\Delta R_A = -m$
3. IF $(\Delta S_1 \oplus \neg \Delta S_2) \vee (S_2 \oplus \neg \Delta S_1)$, $\Delta R_A = +2m$
4. ELSE $\Delta R_A = +0$

We have thus far discussed control logics in terms of perturbations to signals, but this is only a way of simplifying the full control logic for the purpose of discussion.

In fact, we can construct control logics equivalent to the perturbation-based control logics based on the following substitutions. We say that \bullet corresponds to the default value $d_i(c)$ that is set by the control logic to signal i based on the value of the cue c (more explicitly, by the sensor, based on the value of c), with $+$ and $-$ corresponding to any quantity larger than $d_i(c)$ and smaller than $d_i(c)$ respectively. Consider h_i and l_i , which satisfy the inequality $h_i > d_i(c) > l_i$ for all i . We can translate each gate over perturbations into a gate over the signals and the original cue. For $\neg\Delta S_i$ we have

$$g_{-}(S_i, c) = \begin{cases} l_i, & \text{if } S_i > d_i(c) \\ d_i(c), & \text{if } S_i = d_i(c) \\ h_i, & \text{if } S_i < d_i(c) \end{cases}$$

For $\Delta S_1 \wedge \Delta S_2$ (with a ternary AND gate), we have $g_{\wedge}(S_1, S_2, c) = (S_1 > d_1(c)) \wedge (S_2 > d_2(c))$ where \wedge is the ordinary boolean AND gate. For $\Delta S_1 \vee \Delta S_2$, we have $g_{\vee}(S_1, S_2, c) = S_1 + S_2 > d_1(c) + d_2(c)$. For $\Delta S_1 \oplus \Delta S_2$, we have $g_{\oplus}(S_1, S_2, c) = g_{\wedge}(S_1, g_{-}(S_2, c), c) \vee g_{\wedge}(g_{-}(S_1, c), S_2, c)$ where \vee is the ordinary boolean OR gate.

VI. ACKNOWLEDGEMENTS

This work was funded in part by the MIDAS Center for Communicable Disease Dynamics NIH 5U54GM088558-02 and by NSF grant EF-1038590 to CTB, and NIH R01 AI049334 to RA.

References

- [1] R. Dawkins and J.R. Krebs. Arms races between and within species. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 205(1161):489–511, 1979.
- [2] C.T. Bergstrom and R. Antia. How do adaptive immune systems control pathogens while avoiding autoimmunity? *Trends in ecology & evolution*, 21(1):22–28, 2006.
- [3] P. Schmid-Hempel. Variation in immune defence as a question of evolutionary ecology. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1513):357, 2003.
- [4] L. Van Valen. A new evolutionary law. *Evolutionary theory*, 1(1):1–30, 1973.
- [5] J. Nash. Letter to the national security agency. Declassified recently by the NSA, 1955.
- [6] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [7] C.T. Bergstrom and M. Lachmann. The red king effect: when the slowest runner wins the coevolutionary race. *Proceedings of the National Academy of Sciences of the United States of America*, 100(2):593, 2003.
- [8] J.A. Damore and J. Gore. A slowly evolving host moves first in symbiotic interactions. *Evolution*, 2011.

- [9] L.R. Gooding. Virus proteins that counteract host immune defenses. *Cell*, 71(1):5, 1992.
- [10] Peter Palese, editor. *Modulation of host gene expression and innate immunity by viruses*. Springer, 2005.
- [11] F.B. Christiansen, S.P. Otto, A. Bergman, and M.W. Feldman. Waiting with and without recombination: the time to production of a double mutant. *Theoretical population biology*, 53(3):199–215, 1998.
- [12] D.B. Weissman, M.M. Desai, D.S. Fisher, and M.W. Feldman. The rate at which asexual populations cross fitness valleys. *Theoretical population biology*, 75(4):286–300, 2009.
- [13] C.E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59–98, 1949.
- [14] S.C. Kleene, NG de Bruijn, J. de Groot, and A.C. Zaanen. Introduction to metamathematics. 1952.
- [15] R.A. Fisher et al. On the dominance ratio. *Proceedings of the Royal Society of Edinburgh*, 42:321–341, 1922.
- [16] J.B.S. Haldane. A mathematical theory of natural and artificial selection, part v: selection and mutation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 23, pages 838–844. Cambridge Univ Press, 1927.
- [17] RB Campbell. A logistic branching process for population genetics. *Journal of theoretical biology*, 225(2):195–203, 2003.
- [18] F. Gao, L. Han, and K. Schilling. On the rate of convergence of iterated exponentials. *Journal of Applied Mathematics and Computing*, pages 1–8, 2011.
- [19] CP Farrington and AD Grant. The distribution of time to extinction in subcritical branching processes: applications to outbreaks of infectious disease. *Journal of applied probability*, 36(3):771–779, 1999.

Appendix A: Proof of main theorem

We envision the adversary population as evolving by Wright-Fisher dynamics against a controller who deploys a control system that generates a hypercube fitness landscape as sketched in section III. To analyze the learning process of the adversary population, we approximate the Wright-Fisher dynamics using a branching process model. The adversary population begins with a wild type population of individuals that do not perturb the control system at all. From this wild type, mutants arise that perturb one of the control signals up or down; because all single mutants are deleterious, these mutant individuals step down into the fitness valley at some rate per generation rate that is no larger than the product of the population size and the mutation rate $N\mu$.

To cross the valley in k dimensions, the individual needs to found a lineage that accumulates at least k successive mutations and thus this lineage must survive in the fitness valley for at least k generations. Alternatively, a lineage could pick up multiple mutations in each generation—but this won’t help. Consider the case that w mutants arise each time step, and the lineage must survive for at least k/w generations. When k grows very large and w is constant, asymptotic analysis tells us that

the individual needs to found a lineage that survives for at least $\Theta(k)$ generations (because $\lim_{k \rightarrow \infty} |\frac{k/w}{k}| = (1/w)$ and $\lim_{k \rightarrow \infty} |\frac{k}{k/w}| = w$). In this case, generating w -mutants each time-step doesn't speed up crossing the fitness valley appreciably. But what w is allowed to grow with k ? For example, let $w = ck$, where $0 < c \leq 1$. In that case we would only require a lineage to survive for $1/c$ steps, and this is independent of k . But the time that it would take to generate the w mutant required for each successive step is on average $\frac{1}{\mu^{ck}} = \left(\frac{1}{\mu}\right)^{ck}$ where μ is the mutation rate. There is no escape from an average time exponential in k .

It is therefore sufficient to determine how likely an individual who steps down into the valley is to found a lineage that survives in the valley for at least k generations, as k grows very large. If this probability decreases exponentially with k , the average time until we get the first individual who is destined to succeed will increase exponentially in k . To model the fate of such a lineage, we approximate it as a subcritical branching process (15–17). If the relative fitness of an individual in the valley is $0 < \lambda < 1$, this individual will have a Poisson number of offspring with mean λ . Each of these offspring will themselves have Poisson numbers of offspring, again with mean λ . First we prove a technical lemma, and then we show that indeed the average time until we get the first individual who is destined to succeed increases exponentially in k as k grows large.

Lemma 2 $1 - e^{-\lambda} f[e^{\lambda e^{-\lambda}}, k] = |e^{-\lambda} - 1| \lambda^{O(k)}$, where $\lambda < 1$ and $f[b, k]$ is the tetration function $b^{b^{\dots b}}$ with k levels of iterated exponentiation.

Proof First we use the result (Theorem 2 in ref. (18)) that since $\lambda < 1$, $f[e^{\lambda e^{-\lambda}}, k]$ converges to e^λ at a linear rate λ . Thus for any n , we have $|f[e^{\lambda e^{-\lambda}}, n+1] - e^\lambda| = O(\lambda |f[e^{\lambda e^{-\lambda}}, n] - e^\lambda|)$.

From the above, terminating with k and solving the recurrence relation, we obtain:

$$|f[e^{\lambda e^{-\lambda}}, k] - e^\lambda| = \lambda^{O(k)} |f[e^{\lambda e^{-\lambda}}, 0] - e^\lambda|,$$

and so by the definition of tetration, $|f[e^{\lambda e^{-\lambda}}, k] - e^\lambda| = \lambda^{O(k)} |1 - e^\lambda|$. The symmetry property of absolute value implies that $|e^\lambda - f[e^{\lambda e^{-\lambda}}, k]| = \lambda^{O(k)} |1 - e^\lambda|$.

Multiply both sides by $e^{-\lambda}$ to obtain:

$$|1 - e^{-\lambda} f[e^{\lambda e^{-\lambda}}, k]| = \lambda^{O(k)} |e^{-\lambda} - 1|$$

since for $c > 0$, $c|a - b| = |c(a - b)|$. Because $f_k = e^{-\lambda} f[e^{\lambda e^{-\lambda}}, k]$ is the cumulative distribution function (CDF) of some distribution (19), this implies that $0 \leq f_k \leq 1$. Therefore, $0 \leq 1 - f_k \leq 1$, and in particular, $1 - f_k \geq 0$. Consequently we can simplify the absolute value as follows:

$$1 - e^{-\lambda} f[e^{\lambda e^{-\lambda}}, k] = \lambda^{O(k)} |e^{-\lambda} - 1|$$

And thus $1 - e^{-\lambda} f[e^{\lambda e^{-\lambda}}, k] = |e^{-\lambda} - 1| \lambda^{O(k)}$ ■

Lemma 3 As k grows large, the average time until we get the first individual who is destined to succeed assuming its lineage is modeled by a subcritical Poisson branching process (with average number of offspring $0 < \lambda < 1$) is at least $(1/\lambda)^{\Omega(k-1)}/(N\mu)$ generations on average.

Proof By the subcriticality of the branching process that generates the lineage, $\lambda < 1$. The random variable Y is the number of generations the lineage generated by the branching process survives. The CDF of the branching process $f_n = P(Y \leq n)$ gives us the probability of a lineage surviving no more than n generations. Therefore, for our purposes, we need to characterize the probability of non-extinction for k generations, which means we must characterize $P(Y > k-1) = 1 - f_{k-1}$.

By equation 2.5 in Farrington and Grant (19), when $\lambda \leq 1$, $f_n = e^{-\lambda} f[e^{\lambda e^{-\lambda}}, n]$, where $f[b, k]$ is the tetration function. Therefore, by Lemma 2, $P(Y > k-1) = |e^{-\lambda} - 1| \lambda^{O(k-1)}$. The number of mutants who must enter the valley before one does so successfully is on average $1/P(Y > k-1)$, which is thus at least $(1/\lambda)^{\Omega(k-1)}$ since it is easy to show that $0 < |e^{-\lambda} - 1| < 1$. Since the rate at which mutants are produced from the wild type each generation is upper bounded by $N\mu$, the result follows. ■

Now to prove the main Theorem, we combine Lemma 3 with a suitable control logic:

Theorem 1 There exists a control logic on n signals with circuit complexity $O(n)$ is learnable in a number of generations exponential in n .

Proof Without loss of generality we will assume that the controller response with deleterious consequences to the adversaries is R_1 . What this means is that the fitness of the adversary is $1 - \Delta R_1$.

The adversary has three possible actions, upregulate, downregulate, or do nothing. We will now build a control logic that downregulates R_1 only if the adversary upregulates or downregulates each and every one of the signals (rather than doing nothing), and otherwise upregulates R_1 . We will then show that for small enough adversary populations, generating an n -mutant takes time exponential in n .

The control logic L we consider is simply:

1. IF $D(\Delta x_1) \wedge \dots \wedge D(\Delta x_n)$, $\Delta R_1 = +\delta$
2. IF $D(\Delta x_1) \vee \dots \vee D(\Delta x_n)$, $\Delta R_1 = \delta$
3. ELSE $\Delta R_1 = -s$

where $D(x) = (x \Leftrightarrow \bullet)$ and $0 < \delta < 1$. Here \Leftrightarrow represents the logical equivalence operator in Kleene's logic. Generating an optimal type requires a beneficial n -mutant with deleterious intermediates, and by Lemma 3, this takes at least on average $(1/\lambda)^{\Omega(n-1)}/(N\mu)$ generations, for $0 < \lambda < 1$. Since λ is the relative fitness of an individual in the fitness valley $\lambda = 1 - \delta$ by line 1 of L .

Now note that L on the first line has n clauses of the form $\Delta x \Leftrightarrow \bullet$, each of which has 1 gate, and connected by $n-1$ OR gates. In total then there are $n+n-1 = 2n-1$ gates on the first line of L (and this is maximal). The result follows. ■